

REMARKS

This response is submitted in response to a Final Office Action transmitted on December 11, 2007. Claims 1-38 were pending at the time the Office Action was issued. Applicant amends Claims 1-19, 28-29, 35-36, and 38, and cancels Claim 32. Claims 1-31 and 33-38 remain pending.

I. REJECTIONS UNDER 35 U.S.C. § 101

Claims 1-18, 29-36, and 38 are rejected under 35 U.S.C. § 101 as being directed towards non-statutory subject matter. Applicant has amended each of the Claims 1-18, 29-36, and 38 to recite a “computer readable storage medium” instead of a “computer readable storage medium.” Accordingly, Applicant respectfully requests reconsideration and withdrawal of the objections to these claims.

II. REJECTIONS UNDER 35 U.S.C. § 102

Claims 1-2, 4-8, 10, 12-18, 29-31, 33, 36, and 38 are rejected under 35 U.S.C. § 102(e) as having been anticipated by U.S. Patent No. 6,907,572 to Little *et al.* (hereinafter “Little”). Respectfully, Applicant submits that the claims are allowable over the cited reference to Little for at least the reasons explained in detail below.

Claims 1-2, 4-8, 10, and 12-18

Claims 2, 4-8, 10, and 12-18 depend from Claim 1. Claim 1, as amended, recites:

1. A computer readable storage medium encoded with a first data structure, comprising:
 - a parameter definition for at least one input parameter, the parameter definition being configured to enable identification of an appropriate input for the at least one input parameter, wherein the parameter definition is a declared property in a declared class of the first data structure; and
 - an instruction-based mechanism configured to use the parameter definition to identify the appropriate input for the at least one input parameter from information included in an input source, the input source comprising *at least one pipelined live object from a second data structure*,
wherein the instruction-based mechanism is further configured *to apply one or more directives associated with the declared class that specify at least one of a machine role or a user role*, and process the at least one input parameter based on the appropriate input, when the first data structure becomes instantiated into an object. (Emphasis added).

Applicant traverses the rejections. First, Little does not teach or suggest, “the input source comprising *at least one pipelined live object from a second data structure that is identical to the first data structure*,” as recited in Claim 1. (Emphasis added). Instead, Little is silent with respect to the pipelining of objects for use as an input source, including the pipelining of objects from other data structures.

Second, Little also does not teach or suggest, “wherein the instruction-based mechanism is further configured *to apply one or more directives associated with the declared class that specify at least one of a machine role or a user role*,” as recited in Claim 1.

Instead, Little discloses a command line interface (CLI) application that includes XML CLI description files, an XML compiler, a CLI interpreter, as well as classes (e.g., CLIAction class). (Column 4, Lines 34-37; Column 8, Lines 55-56). However, Little does not teach or suggest directives being associated with its classes. Further, since Little does not disclose such directives, Little also does not teach or suggest directives that “*specify at least one of a machine role or a user role,*” as recited in Claim 1.

Accordingly, for at least the reasons stated above, Little does not anticipate Claim 1. Moreover, since Claims 2, 4-8, 10, and 12-18 depend from Claim 1, they are at least allowable due to their dependency, as well as due to additional limitations recited.

Specifically, Claim 7 is further allowable over Little. Claim 7, as amended, recites:

7. The computer readable storage medium of claim 6, wherein the parameter definition comprises a data type and a name for the expected input parameter, and wherein the mechanism further coerces the value having a first data type into a converted value having a second data type specified in the definition.

Applicant respectfully submits that Little does not teach or suggest “wherein the mechanism further coerces the value having a first data type into a converted value *having a second data type specified in the definition.*” (Emphasis added). Instead, Little discloses that if a string input from an argument cannot be parsed, the CLI application “responds to the exception with an output making the invalid argument.” (Column 13, Lines 67 – Column 14, Lines 1; Column 14, Lines 16-19). However, this disclosure of Little does not teach the conversion of the

data type of the value into another data type. Accordingly, Claim 7 is further allowable over Little.

Claims 29-31, 33, 36, and 38

Claims 30-31, 33, 36 and 38 depend from Claim 29. Claim 29, as amended, recites:

29. A computer readable storage medium encoded with a data structure that provides a template for creating an application, the data structure comprising:
- a name identifying an application that is included in a declared parent class provided by an object-based environment;
 - at least one member configured to receive one or more sets of input, wherein each set of input comprises at least one live object; and
 - a method associated with the one or more sets of input, wherein the declared parent class is configured to provide processing that executes the method for each set of input received for the at least one member when the name of the application is invoked, and wherein the application comprises *a command in a pipeline of commands, and the set of input comprises results from a previous command in the pipeline of command;* (Emphasis added).

Applicant traverses the rejections. Specifically, Little does not teach or suggest, “the application comprises *a command in a pipeline of commands, and the set of input comprises results from a previous command in the pipeline of command,*” as recited in Claim 29. (Emphasis added). This limitation is incorporated from canceled Claim 32. Accordingly, as noted in Office Action with respect to Claim 32, Little does not teach or suggest this limitation. (Office Action, Pages 10-11, Paragraph 7).

Moreover, since Claim 32 is rejected under 35 U.S.C. §103(a) as being unpatentable over Little in view of U.S. Patent No. 6,286,035 to Gillis *et al.*

(hereinafter “Gillis”), the distinction between Gillis and the above recited limitation is also hereby discussed. (Office Action, Pages 10-11, Paragraph 7). Notably, the deficiencies of Little with respect to the above recited limitation is not remedied by Gillis. Specifically, Gillis does not teach or suggest a pipeline of commands. The relevant portion of Gillis discloses:

After parsing and validation of the message is accomplished and the data structure is populated, the data structure is sent to another software module on the same embedded system for execution. When the system executes the command and sends the result back to the engine, the data structure containing the output parameters is converted back to the syntax of the operation support system language and sent to the requesting network element as a response to the original command execution request. (Column 4, Lines 65-67; Column 5, Lines 1-5).

In other words, Gillis discloses that once a populated data structure is process by a software module, the data structure containing the “output parameters” is converted and sent to the requesting network element. However, Gillis does not teach or suggest that the “output parameters” are used again as input to populate the data structure. Accordingly, Gillis also does not teach or suggest, “the application comprises *a command in a pipeline of commands, and the set of input comprises results from a previous command in the pipeline of command,*” as recited in Claim 29. (Emphasis added).

Therefore, for at least the reasons stated above, Claim 29 is allowable. Moreover, since Claims 30-31, 33, 36 and 38 depend from Claim 29, they are at least allowable due to their dependency, as well as due to additional limitations recited.

III. REJECTIONS UNDER 35 U.S.C. § 103

Claims 19-23, 26, and 37

Claims 19-23, 26, and 37 are rejected under 35 U.S.C. § 103(a) as being unpatentable over Gillis in view of Little. Claims 20-23, 26 and 37 depend from Claim 19. Claim 19, as amended, recites:

19. A computer-executable method for populating parameters declared within a data structure, the method comprising:
 - obtaining an expected name for a parameter, the expected name being assigned via a declared property in a declared class of a data structure;
 - identifying a label within an input source correlating to the expected name, the input source comprising at least one live object;
 - retrieving a value associated with the label; and
 - assigning the value to the parameter;
 - applying one or more declared directives associated with the declared class of a data structure to the value, the one or more declared directives being associated with the parameter, wherein the one or more declared directives are configured to cause an administrative tool framework to process the parameter.

Applicant traverses the rejections. First, Gillis does not teach or suggest, “applying one or more declared *directives associated with the declared class* of a data structure to the value, the one or more declared directives being associated with the parameter, *wherein the one or more declared directives are configured to cause an administrative tool framework to process the parameter*,” as recited in Claim 19. (Emphasis added). Applicant respectfully submits that this claim element is fully supported under § 112, 1st paragraph, at least by page 38, lines 6-8 and pages 42-43, Table 5 of the Specification. Instead, Gillis discloses a software application that stores command parameters information in data tables.

(Column 5, Lines 40-44). However, Gillis does not teach the use of class and directive declarations.

Second, the deficiencies of Gillis with respect to this limitation are not remedied by Little. Instead, Little discloses a command line interface (CLI) application that includes XML CLI description files, an XML compiler, a CLI interpreter, as well as classes (*e.g.*, CLIAction class). (Column 4, Lines 34-37; Column 8, Lines 55-56). However, Little does not teach or suggest directives being associated with its classes. Further, since Little does not disclose such directives, Little also does not teach or suggest "wherein the one or more declared directives are configured to cause an administrative tool framework to process the parameter," as recited in Claim 19.

Accordingly, the cited references to Little and Gillis, whether individually or in combination, do not teach or suggest the computer-executable method recited in Claim 19. Further, since Claims 20-23, 26, and 37 depend from Claim 19, they are also allowable over the cited references at least due to their dependency, as well as due to additional limitations recited.

Claim 28

Claim 28 is rejected under 35 U.S.C. § 103(a) as being unpatentable over Gillis in view of Little. Claim 28, as amended, recites:

- 28.A system the handles input parameters, the system comprising:
a means for processing; and
a memory means, the memory means being allocated for a plurality of computer-executable instructions which are loaded into the memory means for execution by the means for processing, the computer-executable instructions performing a method comprising:

- a means for obtaining an expected name for a parameter, the expected name being assigned via a declared property in a declared class of a data structure;
- a means for identifying a label within an input source correlating to the expected name, the input source comprising at least one live object;
- a means for retrieving a value associated with the label;
- a means for assigning the value to the parameter; and
- a means for applying one or more declared directives associated with the declared class of a data structure to the value, the one or more declared directives being associated with the parameter, wherein the one or more declared directives are configured to cause an administrative tool framework to process the parameter.

Applicant traverses the rejection. Specifically, Applicant incorporates the reasoning presented above in response to the rejection of Claim 19 under 35 U.S.C. § 103(a), and respectfully submits that that the cited references to Little and Gillis, whether individually or in combination, do not teach or suggest, “a means for applying one or more declared *directives associated with the declared class* of a data structure to the value, the one or more declared directives being associated with the parameter, *wherein the one or more declared directives are configured to cause an administrative tool framework to process the parameter,*” as recited in Claim 28. (Emphasis added).

Claims 32 and 34-35

Claims 32 and 34-35 are rejected under 35 U.S.C. §103(a) as being unpatentable over Little in view of Gillis. Claims 32 and 34-35 depend from Claim 29. Claim 32 is canceled.

Applicant traverses the rejections of Claims 34-35. First, Applicant submits that Little does not teach or suggest, “the application comprises *a command in a pipeline of commands, and the set of input comprises results from a previous*

command in the pipeline of command,” as recited in Claim 29. (Emphasis added). This limitation is incorporated from canceled Claim 32. Accordingly, as noted in Office Action with respect to Claim 32, Little does not teach or suggest this limitation. (Office Action, Pages 10-11, Paragraph 7).

Second, the deficiencies of Little with respect to this limitation are not remedied by Gillis. Specifically, Gillis does not teach or suggest a pipeline of commands. The relevant portion of Gillis discloses:

After parsing and validation of the message is accomplished and the data structure is populated, the data structure is sent to another software module on the same embedded system for execution. When the system executes the command and sends the result back to the engine, the data structure containing the output parameters is converted back to the syntax of the operation support system language and sent to the requesting network element as a response to the original command execution request. (Column 4, Lines 65-67; Column 5, Lines 1-5).

In other words, Gillis discloses that once a populated data structure is processed by a software module, the data structure containing the “output parameters” is converted and sent to the requesting network element. However, Gillis does not teach or suggest that the “output parameters” are used again as input to populate the data structure. Accordingly, Gillis also does not teach or suggest, “the application comprises *a command in a pipeline of commands, and the set of input comprises results from a previous command in the pipeline of command,”* as recited in Claim 29. (Emphasis added).

Further, since Claims 34-35 depend from Claim 29, they are also allowable over the cited references at least due to their dependency, as well as due to additional limitations recited.

Claim 3

Claim 3 is rejected under 35 U.S.C. 103(a) as being unpatentable over Little in view of U.S. Patent No. 6,405,365 to Lee *et al.* (hereinafter "Lee"). Claim 3 depends from Claim 1. Applicant traverses the rejection. First, Applicant incorporates the reasoning presented above in response to the rejection of Claim 1 under 35 U.S.C. § 102(e), and respectfully submits that that Little does not teach or suggest, as recited in Claim 1:

an instruction-based mechanism configured to use the parameter definition to identify the appropriate input for the at least one input parameter from information included in an input source, the input source comprising *at least one pipelined live object from a second data structure that is identical to the first data structure,*

wherein the instruction-based mechanism is further configured *to apply one or more directives associated with the declared class that specify at least one of a machine role or a user role,* and process the at least one input parameter based on the appropriate input to output a live object, when the first data structure becomes instantiated into an object. (Emphasis added).

Moreover, the deficiencies of Little with respect to these limitations are not remedied by Lee. Instead, Lee discloses an Instruction File 150, and generating a Command-Field Record for an instruction "by extracting from the instruction the Command-Field Values specified by the first Syntax Record." (Figure 2; Column 8, Lines 15-20). However, the disclosures of Lee are not relevant to a "pipelined live object" and "applying one or more directives," as recited in Claim 1.

Accordingly, the cited references to Little and Lee, whether individually or in combination, do not teach or suggest the computer readable storage medium

recited in Claim 1. Further, since Claim 3 depends from Claim 1, it is also allowable over the cited references at least due to its dependency, as well as due to additional limitations recited.

Claims 25 and 27

Claims 25 and 27 are rejected under 35 U.S.C. 103(a) as being unpatentable over Gillis in view of Little, and in further view of Lee. Claims 25 and 27 depend from Claim 19.

Applicant traverses the rejections. Specifically, Applicant incorporates the reasoning presented above in response to the rejection of Claim 19 under 35 U.S.C. § 103(a), and respectfully submits that the cited references to Little and Gillis, whether individually or in combination, do not teach or suggest, “applying one or more declared *directives associated with the declared class* of a data structure to the value, the one or more declared directives being associated with the parameter, *wherein the one or more declared directives are configured to cause an administrative tool framework to process the parameter,*” as recited in Claim 19. (Emphasis added).

Moreover, the deficiencies of Gillis with respect to these limitations are not remedied by Lee. Instead, Lee discloses an Instruction File 150, and generating a Command-Field Record for an instruction “by extracting from the instruction the Command-Field Values specified by the first Syntax Record.” (Figure 2; Column 8, Lines 15-20). However, the disclosures of Lee are not relevant to “applying one or more directives,” as recited in Claim 19.

Accordingly, the cited references (Gillis, Little and Lee), whether individually or in combination, do not teach or suggest, “applying one or more declared *directives associated with the declared class* of a data structure to the value, the one or more declared directives being associated with the parameter, *wherein the one or more declared directives are configured to cause an administrative tool framework to process the parameter,*” as recited in Claim 19. (Emphasis added).

Further, since Claims 25 and 27 depend from Claim 19, they are also allowable over the cited references at least due to their dependency, as well as due to additional limitations recited.

Claim 9

Claim 9 is rejected under 35 U.S.C. § 103(a) as being unpatentable over Little in view of Lee, and in further view of Jones (Parse and Validate Command Lines Parameters with VB.NET). Claim 9 depends from Claim 1. Applicant respectfully traverses the rejection. First, Applicant respectfully incorporates the reasoning present above in response to the rejection of Claim 3 under 35 U.S.C. § 103(a). Accordingly, Applicant submits that the cited references to Little and Lee, whether individually or in combination, do not disclose, teach or fairly suggest, as recited in Claim 1:

an instruction-based mechanism configured to use the parameter definition to identify the appropriate input for the at least one input parameter from information included in an input source, the input source comprising *at least one pipelined live object from a second data structure that is identical to the first data structure,*
wherein the instruction-based mechanism is further configured *to apply one or more directives associated with the declared class that specify at least one of a*

machine role or a user role, and process the at least one input parameter based on the appropriate input to output a live object, when the first data structure becomes instantiated into an object. (Emphasis added).

Moreover, the deficiencies of Little with respect to this limitation are not remedied by Jones. Instead, Jones discloses that in VB.Net, a passed command line may be obtained via the Command () function. (Page 2, Paragraph 2, Lines 1-2). However, the disclosures of Jones are not relevant to a “pipelined live object” and “applying one or more directives,” as recited in Claim 1.

Accordingly, the cited references (Gillis, Lee and Jones), whether individually or in combination, do not disclose, teach or fairly suggest the computer readable storage medium recited in Claim 1. Further, since Claim 9 depends from Claim 1, it is also allowable over the cited references at least due to its dependency, as well as due to additional limitations recited.

Claim 11

Claim 11 is rejected under 35 U.S.C. §103(a) as being unpatentable over Little in view of U.S. Patent No. 6,658,625 to Allen *et al.* (hereinafter “Allen”). Claim 11 depends from Claim 1. Applicant respectfully traverses the rejection. First, Applicant respectfully incorporates the reasoning present above in response to the rejection of Claim 1 under 35 U.S.C. § 102(e). Accordingly, Applicant submits that Little does not teach or suggest, as recited in Claim 1:

an instruction-based mechanism configured to use the parameter definition to identify the appropriate input for the at least one input parameter from information included in an input source, the input source comprising *at least one pipelined live object from a second data structure that is identical to the first data structure,*

wherein the instruction-based mechanism is further configured *to apply one or more directives associated*

with the declared class that specify at least one of a machine role or a user role, and process the at least one input parameter based on the appropriate input to output a live object, when the first data structure becomes instantiated into an object. (Emphasis added).

Second, the deficiencies of Little with respect to this limitation are not remedied by Allen. Instead, Allen discloses a parser that is capable of validate and parse an XML document using the Document Type Definition (DTD) of the document. (Column 19, Lines 35-36). Allen also discloses a Program Call Markup Language (PCML) data description that provides configurable data definitions. (Column 6, Lines 35-42). However, the disclosures of Allen are not relevant to a “pipelined live object” and “applying one or more directives,” as recited in Claim 1.

Accordingly, the cited references to Lee and Allen, whether individually or in combination, do not disclose, teach or fairly suggest the computer readable storage medium recited in Claim 1. Further, since Claim 11 depends from Claim 1, it is also allowable over the cited references at least due to its dependency, as well as due to additional limitations recited.

Claim 24

Claim 24 is rejected under 35 U.S.C. §103(a) as being unpatentable over Gillis in view of Little, and in further view of Allen. Claim 24 depends from Claim 19. Applicant respectfully traverses the rejection.

First, Applicant respectfully incorporates the reasoning present above in response to the rejection of Claim 19 under 35 U.S.C. § 103(a). Accordingly, Applicant submits that the cited references to Gillis and Little, whether individually or in combination, do not disclose, teach or fairly suggest, “applying

one or more declared *directives associated with the declared class* of a data structure to the value, the one or more declared directives being associated with the parameter, *wherein the one or more declared directives are configured to cause an administrative tool framework to process the parameter,*” as recited in Claim 19. (Emphasis added).

Second, the deficiencies of Little with respect to this limitation are not remedied by Allen. Instead, Allen discloses a parser that is capable of validate and parse an XML document using the Document Type Definition (DTD) of the document. (Column 19, Lines 35-36). Allen also discloses a Program Call Markup Language (PCML) data description that provides configurable data definitions. (Column 6, Lines 35-42). However, the disclosures of Allen are not relevant to “declared classes” and “declared directives,” as recited in Claim 19.

Accordingly, the cited references (Gillis, Little, and Allen), whether individually or in combination, do not disclose, teach or fairly suggest the computer readable storage medium recited in Claim 19. Further, since Claim 24 depends from Claim 19, it is allowable over the cited references at least due to its dependency, as well as due to additional limitations recited.

In closing, Applicant’s decision not to discuss the differences between the cited art and each dependent claim should not be considered as an admission that Applicant concurs with the conclusions set forth in the Office Action that these dependent claims are not patentable over the disclosure in the cited references. Similarly, Applicant’s decision not to discuss differences between the prior art and every claim element, or every comment set forth in the Office Action, should not be considered as an admission that Applicant concurs with the interpretation and assertions presented in the Office Action regarding those claims. Indeed,


Applicant believes that all of the dependent claims patentably distinguish over the references cited. Moreover, a specific traverse of the rejection of each dependent claim is not required, since dependent claims are patentable for at least the same reasons as the independent claims from which the dependent claims ultimately depend.

CONCLUSION

For the foregoing reasons, Applicant respectfully submits that Claims 1-31 and 33-38 are now in condition for allowance. If there are any remaining matters that may be handled by telephone conference, the Examiner is kindly invited to contact the undersigned attorney at the telephone number listed below.

Respectfully Submitted,

Dated: 4-3-08

By: 
David S. Lee
Reg. No. 38,222

Elliott Y. Chen
Reg. No. 58,293

Lee & Hayes, PLLC
421 W. Riverside Ave, Suite 500
Spokane, WA 99201
Phone: (206) 315-4001 x104
or (206) 315-7914
Fax: (206) 315-4004